

AD-A057 336

MINNESOTA UNIV MINNEAPOLIS MANAGEMENT INFORMATION SY--ETC F/G 9/2
SEARCH MECHANISM ANALYSIS ROUTINE (SMAR): USER'S MANUAL.(U)

JUN 77 D G SEVERANCE, J V CARLIS

N00014-76-C-1063

UNCLASSIFIED

MISRC-TR-77-06

NL

| OF |
AD A057336



END

DATE

FILMED

9-78

DDC

AD A 057336

LEVEL II

10

Report Number: MISRC-TR-77-06

SEARCH MECHANISM ANALYSIS ROUTINE (SMAR):
USER'S MANUAL

Management Information Systems Research Center
Graduate School of Business Administration
University of Minnesota
Minneapolis, MN 55455

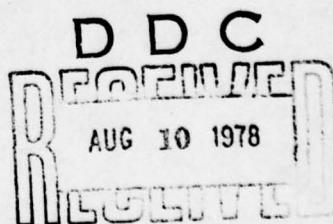
1977 June

AD No. _____
DDC FILE COPY

UNCLASSIFIED; DISTRIBUTION UNLIMITED

Prepared for and delivered to:

David W. Taylor Naval Ship Research and Development Center
ATTN: Code 1821, Dr. David K. Jefferson
U.S. Department of the Navy
Bethesda, MD 20084



The research reported here was funded by the David W. Taylor Naval Ship Research and Development Center under contract N00014-76-C-1063.

78 08 09 070

LEVEL II

1

Report Number: MISRC-TR-77-06

SEARCH MECHANISM ANALYSIS ROUTINE (SMAR):

USER'S MANUAL

John V. Carlis, Research Assistant
Dennis G. Severance, Principal Investigator

Management Information Systems Research Center
Graduate School of Business Administration
University of Minnesota
Minneapolis, MN 55455

1977 June

UNCLASSIFIED; DISTRIBUTION UNLIMITED

Prepared for and delivered to:

David W. Taylor Naval Ship Research and Development Center
ATTN: Code 1821, Dr. David K. Jefferson
U.S. Department of the Navy
Bethesda, MD 20084

D D C
RECORDED
AUG 10 1978
RECORDED
D

The research reported here was funded by the David W. Taylor Naval Ship Research and Development Center under contract N00014-76-C-1063.

78 08 09 070

UNCLASSIFIED, DISTRIBUTION UNLIMITED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9 Final rep. 1 Sep 76-30 Sep 77

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 MISRC-TR-77-06	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Search Mechanism Analysis Routine (SMAR): User's Manual.	5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT 9/1/76 - 9/30/77	
7. AUTHOR(s) 10 Dennis G. Severance / Principal Investigator John V. Carlis	6. PERFORMING ORG. REPORT NUMBER 15 N00014-76-C-1063 / new	
8. PERFORMING ORGANIZATION NAME AND ADDRESS Management Information Systems Research Center 269 19th Ave. So.; University of Minnesota Minneapolis, MN 55455	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 11	
11. CONTROLLING OFFICE NAME AND ADDRESS David W. Taylor Naval Ship Research & Development Center; ATTN: Code 1821, Dr. David K. Jefferson U.S. Department of the Navy; Bethesda, MD 20084	12. REPORT DATE 1977 June 77	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 33	
16. DISTRIBUTION STATEMENT (of this Report)	18. SECURITY CLASS. (of this report) Unclassified 12 320-	
Distribution Unlimited	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited	
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Identifier search mechanisms Record retrieval Hashing identifier search mechanisms	Hashing performance Database design aid	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)	<p>This document is a user's manual for the Search Mechanism Analysis Routine (SMAR) computer program. SMAR is designed to assist a database analyst in the selection of an efficient data record Identifier Search Mechanism (IDSM). To use SMAR a database analyst will enter parameter values describing the records to be stored, the identifiers for these records, and the storage device on which these records will physically reside. Based on these parameters, SMAR will (1) analyze a variety of alternative (over)</p>	

cont
L SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

IDSMS; (2) analyze hashing IDSMS in detail; and (3) compare actual and predicted hashing performance.

ACCESSION FOR	
WTB	White Section <input checked="" type="checkbox"/>
DBR	DBR Section <input type="checkbox"/>
REPRODUCED	<input type="checkbox"/>
JUSTIFICATION	
BY:	
SIGNATURE/AVAILABILITY CODES	
MAIL FILED CDR SPECIAL	
A	

TABLE OF CONTENTS

	<u>Page</u>
1. OVERVIEW	7
1.a Introduction	7
1.b Organization of the User Manual	7
1.c User/Program Menu-Driven Interaction	8
2. ENTER PARAMS	9
2.a Introduction	9
2.b System Params	10
2.b.1 Cell User	10
2.b.2 Noncell User	11
2.c Cost Params	12
2.d Record Params	13
2.e Overflow Method	14
2.f Identifier Params	14
2.g Sample Dialogue to Enter Problem Parameters	15
3. IDSM - IDENTIFIER SEARCH MECHANISM ANALYSIS	18
4. HASHING ANALYSIS	20
4.a Introduction	20
4.b Expected (average) Values	21
4.c Automatic Solution	21
4.c.1 Print Option	22
4.c.2 Solution Parameters	22
4.d User Solution Evaluation	23
4.d.1 Good Divisor	24
4.e Enter Params	24
4.f Sample Terminal Session	25
5. IDSET - IDENTIFIER SET ANALYSIS	29
5.a Introduction	29
5.b Predicted Values	29
5.c Set of Record Identifiers	30
5.d Analysis Results	30
6. OPERATING INSTRUCTIONS	33

SMAR, Search Mechanism Analysis Routine

User Manual

John V. Carlis

and

Dennis G. Severance

1. OVERVIEW

1.a. Introduction

SMAR, Search Mechanism Analysis Routine, is an interactive computer program which aids a system analyst in the design of Identifier Search Mechanisms, IDSMs. Alternative IDSMs exhibit differing performance characteristics when applied to the retrieval of uniquely identified records in a database stored on random access storage devices. SMAR allows an analyst first to enter parameters describing a design problem and then to execute analysis commands which will either: (1) analyze the expected performance of ten different IDSMs, or (2) analyze alternative hashing IDSMs in greater detail, or (3) compare actual versus predicted hashing transformation performance.

This manual explains the usage of SMAR. Where appropriate the user is referred to other literature describing the IDSMs and the analyses in greater detail.

1.b. Organization of the User Manual

The remainder of Section 1 describes the general, menu-driven interaction between the systems and the SMAR programs. Section 2 (PARAMS) defines the set of problem description parameters and the manner in which they are entered into the system. Section 3 (SEARCH MECHANISM) describes the results of evaluating several IDSMs; Section 4 (HASHING) describes the hashing analysis options, and the results printed; Section 5 (ID SET) describes the input of a set of record

identifiers and the comparisons made against predicted performance. Section 6 presents instructions for operating SMAR on a CYBER74 with KRONOS operating system.

1.c. User/Program Menu-Driven Interaction

The user controls the execution of SMAR by selecting commands from a hierachic set of menus. All user inputs to SMAR are numeric and are either a menu choice or a parameter value. The menu choices are of several kinds: perform an analysis, enter one or several parameters, list parameters, go to a higher menu, go to a lower menu, or stop (terminate the program). After each entry the user is presented with another menu as illustrated by the following dialogue:

```
ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
            3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 1
ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 3
ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
            4=LIST SPECS,5=BRIEF ENTRY,6=MFNU,7=TOP

? 5
ENTER, IN ORDER: RECORDSIZE.NO.OF RECS,RFT/REC/MO

? 39000 900 60
ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
            4=LIST SPECS,5=BRIEF ENTRY,6=MFNU,7=TOP

? 6
ENTER PARAM CODE. 1=SYSTFM,2=COST,3=RECORD,4=OVFRFLW,5=ID,6=MENU
```

In this dialogue the user is initially presented with the top level, or SMAR, menu. Before entering a response the user must wait for the prompt symbol '?'. The user enters a '1' and the program responds with the 'PARAM' level menu. The user enters a '3'; the program responds with the 'RECORD' level menu. Since

all three record params (in brief mode) are to be input, the user enters a '5' ... and so on. The dialogue will continue until the user enters '5' (for 'stop') when presented with the SMAR menu.

The user might occassionally enter incorrect or incomplete responses. The program responds to a blank line (or to insufficient entries in a brief entry mode) with an additional prompt, '?', for the remaining data. If an invalid menu code is entered, the user is asked to re-enter the value.

The following sections describe user menus obtained via the response codes to the top level SMAR menu.

**ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP**

2. ENTER PARAMS (SMAR response code : 1)

2.a. Introduction

In response to the selection of ENTER PARAMS a menu with five different parameter types is presented:

ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

These parameters describe either the computer system used (SYSTEM and COST), the records to be stored (RECORD and ID), or the solution method for hashing analysis (OVERFLOW). Not all parameters are applicable for all analyses. SYSTEM, COST, and RECORD are always required, while OVERFLOW only applies to HASHING and KEY only applies to IDSM and IDSET.

For each parameter type the choices are:

- enter a single parameter value
- list the current values
- enter all values in a brief mode
- return to the next higher level menu ('menu')
- return to the menu above the enter param level ('top')

Sections 2.b through 2.f describe the five parameters types and Section 2.g presents a typical user dialogue.

2.b System Params (Params response code: 1)

System params first must be entered for all problem analyses. The menu presented is:

SYSTEM MENU: ENTER CODE. 1=CELL USER,2=NONCELL USFR,3=LIST SPECS,4=MENU,5=TOP

A user can design search mechanisms for either of two types of secondary memory: cell or noncell. These memories have differing parameter operating characteristics. The IDSM routine always assumes a cell environment while the HASHING routine will analyze either environment; IDSET makes no direct use of system parameters.

A response code of 1 declares that a cell environment is to be used and presents the cell menu (see below), while a code of 2 declares a noncell environment and invokes the noncell menu. Code 3 lists the current declaration; code 4 invokes the PARAMS menu, and code 5 presents the SMAR menu.

2.b.1. Cell User (System Params response code: 1)

**ENTER CODE. 1=CELLSIZE,2=PTRSIZE,3=CLOC,4=CRAN,
5=LIST SPECS,6=BRIEF ENTRY,7=MENU,9=TOP**

1. cellsize

the length in bytes (i.e., characters) of a secondary memory cell; that is the collection of data transferred from secondary memory as a single unit (e.g., track, sector, page, block, physical record).

**ENTER CELLSIZE
? 4000**

2. ptrsize

the length in bytes of a memory pointer.

**ENTER PTRSIZE
? 4**

3. CLOC

the time in milliseconds for a local memory access. This value includes the time to transfer the cell.

**ENTER CLOC
? 40**

4. CRAN
ENTER CRAN
? 70 the time in milliseconds for a random memory access. This value includes the time to transfer the cell.

5. list specs list the current cell user specifications.
CELL USER SPECS:CELLSIZE= 4000. ,PTRSIZE= 4.
CLOC(MSEC)= 40.00 , CRAN(MSEC)= 70.00

6. brief entry enter values for cell size, ptrsize, CLOC and CRAN in that order.
ENTER, IN ORDER:CELLSIZE,PTRSIZE,CLOC,CRAN.
? 4000 4 40 70

7. menu

8. top

2.b.2. Noncell User (System Params response code: 2)

**ENTER CODE. 1=TRKSIZE,2=GAP SIZE,3=PTR SIZE,4=TLOC,5=TRAN,6=TFRATE,
 7=SPAN CODE ,8=LIST SPECS,9=BRIEF ENTRY,10=MENU,11=TOP**

1. tracksize the length in bytes of a track.
ENTER TRKSIZE
? 13000

2. gapsize the length in bytes of the gap between blocks (i.e., physical records) on a track.
ENTER GAPSIZE
? 135

3. ptrsize the length in bytes of a pointer.
ENTER PTRSIZE
? 4

4. TLOC the time in milliseconds for a local access to a block.
ENTER TLOC(MSEC)
? 10.2

5. TRAN the time in milliseconds for a random access to a block.
ENTER TRAN(MSEC)
? 15.7

6. TFRATE the transfer rate of a channel device combination in K bytes/SEC (e.g., a channel speed of 806,000 bytes/sec. should be entered as 806).
ENTER TFRATE(K)
? 806

7. span code indicates whether or not a block can span track boundaries.

ENTER CODE. 1=NO SPANNED TRACKS, 2=SPANNED TRACKS.

? 1

8. list specs list the current noncell user values.

NONCELL USER SPECS-TRACK(BYTES)= 13000. GAP(BYTES)= 135. PTR(BYTES)= 4
TLOC(MSEC)= 10.20 TRAN(MSEC)= 17. TRATE(KBYTES/SFC)= 806.
AND NO SPANNED TRACKS

ENTER, IN ORDER: TRKSIZE, GAP SIZE, PTR SIZF, TLOC, TRAN, TFRATE, SPAN CODE

? 13000,135,4,10.2,16.7,806,1

10. menu

11. top

2.c. Cost Params (Params response code: 2)

Cost parameters must be entered for all analyses.

ENTER CODE. 1=CT,2=CA,3=CS,4=LIST SPECS,5=BRIEF ENTRY,6=MFNU,7=TOP

where C_T = cost of transfer time, (measured in \$/second), describes the holding costs on such resources as main memory, the channel, the device controller and, of course, the user.

C_A = cost of access time (in \$/second). This cost may be less than C_T if system resources such as the channel or device control unit are released during an access.

C_S = cost of storage (\$/track/time-period).

For a cell user:

- C_T is not used; the cost of transfer is reflected in the access a record (the cell size is constant).
- C_S is \$/cell/time period.

Codes: 1-3 enter CT, CA, or CS individually

ENTER CT ENTER CA ENTER CS
? .01 ? .01 ? .2

4. list specs

COST SPECS- CT,COST OF TRANSFER TIME IN \$/SEC= .010
CA,COST OF ACCESS TIME IN \$/SEC= .010
CS,COST OF STORAGE IN \$/TRK/MO= .200

5. brief entry

ENTER, IN ORDER: CT,CA,CS
? .01,.01,.20

6. menu

7. top

2.d. Record Params (Params response code: 3)

Record parameters must be entered for all analyses.

ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RET/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

1. record size - the length in bytes.

ENTER RECORDSIZE
? 990

2. no. of records - count of logical records to be stored in the database.

ENTER NO.OF RECS
? 22500

3. RET/REC/MO - the intensity of retrieval against the file measured by the number of retrievals per record per month. For example with 10,000 records and RET/REC/MO = 10 there are 100,000 retrievals in a month.

ENTER RFT/REC/MO
? 35

4. list specs - list the current stored values.

RECORD SPECS- RECORD SIZE= 990. NO.OF RECORDS= 22500.
RETRIEVALS/RECORD/MO= 35.000

5. brief entry - enter values for record size, no. of records and retrieval intensity in that order.

ENTER, IN ORDER: RECORDSIZE,NO.OF RECS,RET/REC/MO
? 990,22500,35

6. menu
7. top

2.e. Overflow Method (Params response code: 4)

Overflow method applies only to hashing analysis.

ENTER CODE. 1=OPEN,2=CHAINBD,3=LIST VALUE,4=MENU,5=TOP

1. open open overflow method.
2. chained chained overflow method with a separate overflow area. For noncell users overflow bucketsize is one; for cell users overflow bucksize is the maximum number of records which fit in a cell with a chain pointer.
3. list specs list the current value
OVERFLOW METHOD IS OPEN
OR
OVERFLOW METHOD IS CHAINED
4. menu
5. top

2.f. Identifier Params (Params response code: 5)

Parameters describing the set of record identifiers must be entered before executing either IDSM or IDSET.

**ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP**

1. ID length the number of bytes in the identifier field.
ENTER ID LENGTH
? 5
2. ID position the displacement of the identifier field from the beginning of a stored record.
ENTER ID STARTING POSITION
? 1
3. ID type the vocabulary of symbols used to compose an identifier.
ENTER ID TYPE CODE. 1=NUMERIC,2=ALPHA,3=ALPHANUMERIC
? 1

The number of possible unique identifiers, call vocabulary size, is a function of the identifier's length and type:

X^{ID} length where X is 10, 26, or 36.

2.g. Sample Dialogue to Enter Problem Parameters

This section presents a sample session for entering design problem parameter values. Some of the terminal entries have been annotated.

start the program

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 1
ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

1

SYSTEM MENU: ENTER CODE. 1=CELL USER,2=NONCELL USER,3=LIST SPECS,4=MENU,5=TOP

? 3

look at the default values

NONCELL USER SPECS-TRACK(BYTES)= 13030. GAP(BYTES)= 135. PTR/BYTES)= 4.
TLOC(MSEC)= 10.00 TRAV(MSEC)= 16.72 FRATE(KBYTES/SEC)= 826.
AND NO SPANNED TRACKS

SYSTEM MENU: ENTER CODE. 1=CELL USER,2=NONCELL USER,3=LIST SPECS,4=MENU,5=TOP

1

choose to be a cell use

ENTER CODE. 1=CELLSIZE,2=PTRSIZE,3=LOC,4=CRAN,
5=LIST SPECS,6=BRIEF ENTRY,7=MENU,8=TOP

? 6

ENTER, IN ORDER: CELLSIZE, PTRSIZE, CLOC, CRAN.

? 4000 40 70 4

User notices incorrect order and will re-enter 6 in response to System params menu.

ENTER CODE. 1=CELLSIZE, 2=PTRSIZE, 3=CLOC, 4=CRAN,
5=LIST SPECS, 6=BRIEF ENTRY, 7=MENU, 8=TOP

? 6

ENTER, IN ORDER: CELLSIZE, PTRSIZE, CLOC, CRAN.

? 4000 4 40 70

ENTER CODE. 1=CELLSIZE, 2=PTRSIZE, 3=CLOC, 4=CRAN,
5=LIST SPECS, 6=BRIEF ENTRY, 7=MENU, 8=TOP

? 7

SYSTEM MENU: ENTER CODE. 1=CELL USER, 2=NONCELL USER, 3=LIST SPECS, 4=MENU, 5=TOP

? 4

ENTER PARAM CODE. 1=SYSTEM, 2=COST, 3=RECORD, 4=OVERFLOW, 5=ID, 6=MENU

? 2

ENTER CODE. 1=CT, 2=CA, 3=CS, 4=LIST SPECS, 5=BRIEF ENTRY, 6=MENU, 7=TOP

? 3

ENTER CS

? .20

ENTER CODE. 1=CT, 2=CA, 3=CS, 4=LIST SPECS, 5=BRIEF ENTRY, 6=MENU, 7=TOP

? 6

ENTER PARAM CODE. 1=SYSTEM, 2=COST, 3=RECORD, 4=OVERFLOW, 5=ID, 6=MENU

? 3

ENTER CODE. 1=RECORD SIZE, 2=NO.OF RECORDS, 3=RET/REC/MO,
4=LIST SPECS, 5=BRIEF ENTRY, 6=MENU, 7=TOP

? 4

Look at default values

RECORD SPECS- RECORD SIZE= 1000. NO.OF RECORDS= 20000.
RETRIEVALS/RECORD/MO= 10.000

ENTER CODE. 1=RECORD SIZE, 2=NO.OF RECORDS, 3=RET/REC/MO,
4=LIST SPECS, 5=BRIEF ENTRY, 6=MENU, 7=TOP

? a blank line is entered -- another ? appears.
? 9 9 is an invalid menu choice.

9 IS INVALID -- REENTER
ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 5

ENTER, IN ORDER: RECORDSIZE,NO.OF RECS,RFT/REC/MO

brief entry can take more than one line

? 950
? 10000 50
ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 6
ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 4

ENTER CODE. 1=OPEN,2=CHAINED,3=LIST VALUE,4=MENU,5=TOP

? 2
ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 5

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 1

ENTER ID LENGTH

? 1 *DEL*

5

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 2

ENTER ID STARTING POSITION

? 1

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 3

ENTER ID TYPE CODE. 1=NUMERIC,2=ALPHA,3=ALPHANUMERIC

? 1

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 4
ID SPECS: ID LENGTH= 5. ,STARTING POSITION= 1 ,ID TYPE= NUMERIC

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIFF ENTRY,6=MENU,7=TOP

? 7

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

3. IDSM - IDENTIFIER SEARCH MECHANISM ANALYSIS (SMAR response code: 2)

A system designer is faced with selecting from a number of available identifier search mechanisms (IDSMs). The purpose of the search mechanism analysis portion of SMAR is to assist the designer in comparing alternative IDSMs for a cell memory environment. The output from IDSM is in table form. For each of ten IDSMs the table contains values for five measures: the average number of identifier comparisons made for one retrieval, the search time in MSEC for that retrieval, the cost in dollars for that retrieval, the number of cells required to store the records (plus any associated indices and overflow areas), and the percentage space overhead.

For details of the IDSMs the reader is referred to: (1) D. G. Severance, "Record Retrieval Via Hashing Functions and Scatter Tables," in National Bureau of Standards Software Engineering Handbook, January, 1977; and (2) J. Martin, Computer Data-Base Organization, Prentice-Hall, 1975.

An IDSM analysis has no additional user dialogue. After the table is printed the SMAR menu is presented. In the following example RECORD and ID parameters are entered and IDSM makes use of default CELL and COST parameters.

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 1

ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU,

? 3

ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RET/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 5

ENTER, IN ORDER: RECORDSIZE,NO.OF RECS,RFT/REC/MO

? 1000 22500 35

ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RET/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 6

ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 5

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 5

ENTER, IN ORDER: ID LENGTH, STARTING POSITION, TYPE CODE

? 5 1 1

ENTER CODE. 1=ID LENGTH,2=ID POSITION,3=ID TYPE,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 7

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 2

RECORD LENGTH 1000. NO. OF RECORDS 22500. RFT/REC/MO 35.000
ID LENGTH 5. VOCABULARY SIZE 1000+36

PERFORMANCE CHARACTERISTICS FOR COMMON RECORD SEARCH TECHNIQUES

TECHNIQUE	NUMBER OF COMPARISONS	SEARCH TIME (MSEC)	CELLS REQUIRED	SPACE OVERHEAD	ACCESS COSTS SINGLE, ACCESS
DIRECT ADDRESSING	1.00	20.00	25202	344.44	.000700
SEQUENTIAL SCAN	11250.50	112525.00	5625	0	1.125250
BINARY SEARCH					
TRIE	20.24	1416.35	7520	33.33	.014163
BY CELL	13.96	822.03	5625	0	.003020
ON FULL INDEX	227.17	397.07	5676	.01	.003971
INDEX					
FULL	11252.52	1142.00	5676	.01	.011420
CELL	2815.50	773.78	5678	.27	.007734
HIERARCHIC CELL	231.37	210.00	5679	.25	.002107
HASHING					
OPEN OVERFLOW	4.42	22.29	5665	18.50	.002323
CHAINED OVERFLOW	2.37	110.30	5452	50.23	.001103

4. HASHING ANALYSIS (SMAR response code: 3)

4.a. Introduction

HASHING ANALYSIS will aid in the selection of design parameters for a hashing search mechanism. This design aid is based upon a mathematical model presented in Severance and Duhne, "A Practitioner's Guide to Addressing Algorithms," Comm ACM (June 1975), pages 314-326. The user is assumed to be familiar with that article.

Three types of parameters are of interest -- computer, record, and solution. Computer parameters describe the secondary storage device performance and cost characteristics. Record parameters describe the data file in terms of record size, number of records, and intensity of record retrieval. The three essential solution parameters are:

- the bucketsize, S (the number of records per bucket);
- the loading factor, P (the number of records to be stored divided by the number of record slots allocated in the prime storage area); and
- the bucket overflow method (open or chained to a separate overflow area).

The hashing analysis menus provides several options. Following the execution of an analysis this menu is presented again.

**ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USFR SOLN EVALUATED,
4=ENTER PARAMS,5=MENU**

These options are described briefly here and in more detail in Sections 4.b-4.f.

1. expected values	the expected number of accesses and, only with chained overflow, the expected overflow per bucket are calculated, for a bucket size, loading factor and overflow method (assuming a uniformly distributed transformation values).
2. automatic solution	the program will find the solution parameters S and P , for a given overflow method, which yield the lowest cost.

- 3. user solution evaluation the user enters values for S, P, and overflow method. The program calculates the resulting cost.
- 4. enter params return to PARAMS menu.
- 5. menu return to the SMAR menu.

The following sections explain each of the menu choices and the resulting output. Section 9 presents a sample session.

4.b. Expected (average) Values (HASHING response code: 1)

The expected value calculations follow those in Severance and Duhne. The user is prompted to enter a bucket size, loading factor, and overflow method.

EXPECTED VALUE SETUP:ENTER BUCKET SIZE AND LOAD FACTOR

? 8 , 8

ENTER CODE. 1=OPEN, 2=CHAINED, 3=LIST VALUE, 4=MENU, 5=TOP

1

The expected number of accesses required to retrieve a record is printed for overflow.

FOR BUCKET SIZE= 8. AND LOAD FACTOR= .8000
E(ACCESES)= 1.152

If chained overflow were selected, then both the expected number of accesses and the expected overflow (# records) per bucket would be given. The latter value is useful in calculating the amount of overflow space needed.

FOR BUCKET SIZE= 8. AND LOAD FACTOR= .8000
E(OVERFLOW)= .429 AND E(ACCESSFS)= 1.136

4.c. Automatic Solution (HASHING response code: 2)

The automatic solution portion of the program selects the best (minimal cost) values for bucket size and load factor, for a given overflow method. The resulting output, as described below, consists of the problem and solution parameters plus some derived statistics. The solution method used is explained in

detail in the Severance and Duhne article. Briefly, for each reasonable bucket size* the loading factor which yields the lowest cost solution is found via a quadratic search routine. The best solution over all the bucket sizes is selected as optimal.

4.c.1. Print Option

ENTER PRINT OPTION. 1=FINAL RESULTS ONLY, 2= ALL
? 1

1. final results only print only the optimal solution.
2. all print the minimal solution for each reasonable bucketsize.

The all option could be useful to the user if, for example, there was little difference in cost between the optimal solution and another solution with a smaller bucketsize. If buffer space was at a premium then perhaps the user would choose to implement the latter solution.

4.c.2. Solution Parameters

This section explains the format of an output solution such as the one below.

***** MINIMAL SOLUTION *****

OVERFLOW METHOD	CHAINED	LOAD FACTOR, P	.68
BUCKET SIZE, S	3.	#BUCKETS	189.
#RECORDS	1000.	RETR/REC/MO	1000.000
RECORD LENGTH	1000.		
AVERAGE OVERFLOW/BUCKET	.23		
AVERAGE #ACCESSES TO A RECORD	1.17		
TOTAL COST	327.65		
COST OF TIME	234.79		
COST OF SPACE	92.87		
TIME IN SECS	23479.		
SPACE IN TRACKS	133.		
TIME/RETRIEVAL IN MSEC	23.5	COST/RETRIEVAL	.000235

* For noncell users reasonable bucket sizes will avoid large amounts of track waste in the primary area. Only bucket sizes of 1 are considered for the overflow area. For cell users the only reasonable bucket size for the primary area is the maximum that will fit in a single cell.

The solution table contains:

1. Solution parameters: overflow method, bucket size and loading factor.
2. Problem parameters: number of records, record length, and retrieval intensity.
3. Expected values: average overflow per bucket (chained only) and average number of accesses to a record.
4. Computer dependent expected values: number of MSEC per retrieval and the cost per retrieval.
5. Monthly costs for the solution: total cost and its components -- cost for space, cost for time.
6. Time and space for the solution: number of seconds per month
number of storage units -- cells or tracks

4.d. User Solution Evaluation (HASHING response code: 3)

A user is permitted to define an arbitrary solution (bucketsize, loading factor, and overflow method) produced by the automatic solution; the only difference is in the table title. He is prompted as follows:

USER SOLN SETUP

**ENTER CODE. 1=BUCKETSIZE,2=LOADING FACTOR,3=OVERFLOW,
4=EVALUATE USER SOLN,5=MENU**

The user may change any of the current solution parameters.

1. bucketsize

ENTER BUCKETSIZE

? 12

2. loading factor

ENTER LOAD FACTOR

.68 .

3. overflow method

ENTER CODE. 1=OPEN,2=CHAINED,3=LIST VALUE,4=MENU,5=TOP

? 2

4. evaluate user soln

evaluate the solution and print a summary table as shown below.

***** USFR SOLN EVALUATION *****

OVERFLOW METHOD	CHAINED	LOAD FACTOR, P	.58
BUCKET SIZE, S	12.	#BUCKETS	123.
#RECORDS	1200.	RETR/RFC/MO	1020.000
RECORD LENGTH	1200.		
AVERAGE OVERFLOW/BUCKET	.15		
AVERAGE #ACCESSES TO A RECORD	1.01		
TOTAL COST	750.95		
COST OF TIME		725.27	
COST OF SPACE		25.72	
TIME IN SEC'S	72523.		
SPACE IN CELLS	129.		
TIME/RETRIEVAL IN MSFC	72.5	COST/RETRIEVAL	.000725

4.d.1. Good Divisor

After the solution is evaluated, the user is asked if a similar but theoretically better divisor should be computed.

GOOD HASHING DIVISOR DESIRED? 1=YES,2=NO

? 1

When using division hashing as a transformation technique, some divisions will theoretically produce a more uniform distribution of bucket addresses from a set of keys. See Severance and Duhne, page 315 for an explanation.

1. yes

the divisor returned is greater than the existing number of buckets and contains no factor less than 20. The solution is re-evaluated and the solution table printed.

2. no

the hashing menu is presented.

4.e. Enter Params (HASHING response code: 4)

Between problem analyses the user may choose to change parameter values by returning to the PARAM menu. See Section 2.

4.f. Sample Terminal Session

/-SMAR

? 1 ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 1 ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 3 ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 5

ENTER, IN ORDER: RECORDSIZE,NO.OF RECS,RFT/REC/MO

? 950,10000,50
ENTER CODE. 1=RECORD SIZE,2=NO.OF RECORDS,3=RFT/REC/MO,
4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 6 ENTER PARAM CODE. 1=SYSTFM,2=COST,3=RECORD,4=OVFRFLOW,5=ID,6=MENU

? 1

SYSTEM MENU: ENTER CODE. 1=CELL USFR,2=NONCELL USER,3=LIST SPECS,4=MENU,5=TOP

? 3

NONCELL USER SPECS-TRACK(BYTES)= 13030. GAP(BYTES)= 135. PTR(BYTES)= 4.
TLOC(MSEC)= 10.00 TRAN(MSEC)= 17. TFRATE(KBYTES/SEC)= 806.
AND NO SPANNED TRACKS

SYSTEM MENU: ENTER CODE. 1=CELL USER,2=NONCELL USER,3=LIST SPECS,4=MENU,5=TOP

? 4 ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 2 ENTER CODE. 1=CT,2=CA,3=CS,4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 4

COST SPECS- CT,COST OF TRANSFER TIME IN \$/SEC= .010

CA,COST OF ACCESS TIME IN \$/SEC= .010

CS,COST OF STORAGE IN \$/TRK/MO= .700

ENTER CODE. 1=CT,2=CA,3=CS,4=LIST SPECS,5=BRIEF ENTRY,6=MENU,7=TOP

? 7

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 3
ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USER SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 1

EXPECTED VALUE SETUP:ENTER BUCKET SIZE AND LOAD FACTOR

? 4 1

ENTER CODE. 1=OPEN,2=CHAINED,3=LIST VALUE,4=MENU,5=TOP

? 2

FOR BUCKET SIZE= 4. AND LOAD FACTOR= 1.0000
E(OVERFLOW)= .781 AND E(ACCESES)= 1.391

ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USER SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 1

EXPECTED VALUE SETUP:ENTER BUCKET SIZE AND LOAD FACTOR

? 12 1

ENTER CODE. 1=OPEN,2=CHAINED,3=LIST VALUE,4=MEVU,5=TOP

? 2

FOR BUCKET SIZE= 12. AND LOAD FACTOR= 1.0000
E(OVERFLOW)= 1.372 AND E(ACCESES)= 1.326

ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USER SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 2

ENTER PRINT OPTION. 1=FINAL RESULTS ONLY,2= ALL

? 1

***** MINIMAL SOLUTION *****

OVERFLOW METHOD	CHAINED	LOAD FACTOR, P	1.07
BUCKET SIZE, S.	13.	#BUCKETS	720.
#RECORDS	10000.	RETR/REC/MO	50.000
RECORD LENGTH	950.		
AVERAGE OVERFLOW/BUCKET	1.95		
AVERAGE #ACCESSES TO A RECORD	1.45		
TOTAL COST	793.46		
COST OF TIME	200.31		
COST OF SPACE	593.16		
TIME IN SECS	20021.		
SPACE IN TRACKS	847.		
TIME/RETRIEVAL IN MSEC	40.1	COST/RETRIEVAL	.000101

GOOD HASHING DIVISOR DESIRED? 1=YES,2=NO

? 2

ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USFR SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 4

ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 4

ENTER CODE. 1=OPEN,2=CHAINED,3=LIST VALUE,4=MENU,5=TOP

? 1

ENTER PARAM CODE. 1=SYSTEM,2=COST,3=RECORD,4=OVERFLOW,5=ID,6=MENU

? 6

ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USFR SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 2

ENTER PRINT OPTION. 1=FINAL RESULTS ONLY,2= ALL

? 1

***** MINIMAL SOLUTION *****

OVERFLOW METHOD	OPEN		
BUCKET SIZE, S	13.	LOAD FACTOR, P	.91
#RECORDS	10000.	#BUCKETS	841.
RECORD LENGTH	950.	RETR/REC/MO	50.000
AVERAGE #ACCESSES TO A RECORD	1.31		
TOTAL COST	788.51		
COST OF TIME		199.68	
COST OF SPACE		588.83	
TIME IN SECS	19968.		
SPACE IN TRACKS	841.		
TIME/RETRIEVAL IN MSEC	39.9	COST/RETRIEVAL	.000399

GOOD HASHING DIVISOR DESIRED? 1=YES,2=NO

? 1

OVERFLOW METHOD	OPEN		
BUCKET SIZE, S	13.	LOAD FACTOR, P	.90
#RECORDS	10000.	#BUCKETS	851.
RECORD LENGTH	950.	RETR/REC/MO	50.000
AVERAGE #ACCESSES TO A RECORD	1.26		
TOTAL COST	789.35		
COST OF TIME		193.65	
COST OF SPACE		595.70	
TIME IN SECS	19365.		
SPACE IN TRACKS	851.		
TIME/RETRIEVAL IN MSEC	39.7	COST/RETRIEVAL	.000397

ENTER CODE. 1=EXPECTED VALUFS,2=AUTOMATED SOLN,3=USER SOLN EVALUATED.
4=ENTER PARAMS,5=MENU

? 3

USER SOLN SETUP

ENTER CODE. 1=BUCKETSIZE,2=LOADING FACTOR,3=OVERFLOW,
4=EVALUATE USER SOLN,5=MENU

? 1

ENTER BUCKETSIZE

? 4

ENTER CODE. 1=BUCKETSIZE,2=LOADING FACTOR,3=OVERFLOW,
4=EVALUATE USER SOLN,5=MENU

? 4

***** USER SOLN EVALUATION *****

OVERFLOW METHOD	OPEN		
BUCKET SIZE, S	4.	LOAD FACTOR, P	.90
#RECORDS	10000.	#BUCKETS	2766.
RECORD LENGTH	950.	RETR/REC/MO	50.000
AVERAGE #ACCESSES TO A RECORD	2.05		
TOTAL COST	929.63		
COST OF TIME	184.34		
COST OF SPACE	645.34		
TIME IN SEC'S	18434.		
SPACE IN TRACKS	922.		
TIME/RETRIEVAL IN MSEC	36.9	COST/RETRIEVAL	.000369

GOOD HASHING DIVISOR DESIRED? 1=YES,2=NO

? 2

ENTER CODE. 1=EXPECTED VALUES,2=AUTOMATED SOLN,3=USER SOLN EVALUATED,
4=ENTER PARAMS,5=MENU

? 5

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

? 5

STOP

5. IDSET - IDENTIFIER SET ANALYSIS (SMAR response code: 4)

5.a. Introduction

The purpose of IDSET is to compare the predicted mapping of records into a hashing file versus the actual mapping generated by transforming a set of record identifiers. In predicting system performance the hashing analysis assumes that the transformation produces randomly distributed bucket addresses. The user can test whether actual identifiers when transformed approximate a random distribution or are subject to clustering (clustering degrades system performance).

The transformation used is division by the number of prime buckets with the remainder used as the bucket address. Hashing analysis generates the number of prime buckets and should be executed before IDSET.

The predicted values are explained in Section b; Section c explains how to input and describe the set of identifiers; and Section d explains the statistics generated and presents an example.

5.b. Predicted Values

Severance and Duhne, page 315, state:

A good transformation is one which randomizes the assignment of identifiers over the entire memory space.

Precisely, one's objective is to construct a transformation such that a randomly selected identifier has an equal chance of mapping to any bucket. Probabilistically, this situation is described by the uniform distribution $P(\text{assigned bucket} = k) = 1/b$ for $k = 1, 2, \dots, b$.

This randomizing rule of thumb has achieved popularity for two reasons. First, the expected performance of such a transformation is easily analyzed and, second, this performance is surprisingly good. When the rule is satisfied, the distribution of the number of records per bucket, X , is binomial,

$$P_{N,b}(X = i) = \binom{N}{i} (1/b)^i ((b-1)/b)^{N-i}$$

and accurately approximated by the Poisson density function

$$P_m(X = i) = e^{-m} m^i / i!$$

This Poisson density function is used to calculate the expected number of buckets receiving X number of records for X from 0 to $2M$ (M = bucketsize times loading factor). Also calculated is the number of records expected to overflow.

5.c. Set of Record Identifiers

The actual loading of records into buckets is generated from the transformation of record identifiers into bucket addresses. The user must enter parameter values for the ID length, position, and type via PARAMS before invoking IDSET and the number of identifiers read is equal to the number of records defined to PARAMS.

Section 6 explains the KRONOS control cards required to input the file of identifiers.

5.d. Analysis Results

To reasonably compare the predicted and actual mappings, the number of records read (one identifier per record) should equal the number of records described to the HASHING module. IDSET reads only that number of records, and ignores any excess.

As shown below, the output from IDSET summarizes: the number of prime buckets, loading factor, bucket size, number of records read, acutal count of records which overflow, and the number which were expected to overflow. Actual overflow much larger than expected means that actual system cost will be higher than predicted.

A summary table which compares the actual and predicted mappings is then generated. With a reasonable transformation few buckets have more than twice the bucket size times loading factor (or $2*M$) records mapped to them. Therefore, there are $2*M$ rows in the table. Printed in each row is a record count, the actual and predicted number of buckets with that count, and the difference

between these two values. As an example, for the count 16, 4 buckets had this number of records mapped; 3.13 buckets were predicted, so the actual number is .87 more than predicted (a small difference).

IDENTIFIER SET ANALYSIS

NUMBER OF BUCKETS 39 LOADING FACTOR 1.87 SLOTS PER BUCKET 12.
NUMBER OF RECORDS READ 2000

ACTUAL OVERFLOW 932. EXPECTED OVERFLOW 933.

COUNT OF BUCKETS THAT RECEIVED X RECORDS		NO RECORDS	ACTUAL	EXPECTED	DIFFERENCE
0	1				
0	0	0	.0000	-.0000	
1	0	0	.0000	-.0000	
2	0	0	.0000	-.0000	
3	0	0	.0000	-.0000	
4	0	0	.0002	-.0002	
5	0	0	.0007	-.0007	
6	0	0	.0028	-.0028	
7	0	0	.0089	-.0089	
8	0	0	.0250	-.0250	
9	0	0	.0624	-.0624	
10	0	0	.1402	-.1402	
11	0	0	.2963	-.2963	
12	0	0	.5352	-.5362	
13	1.	1.	.9268	.0732	
14	1.	1.	1.4877	-.4877	
15	0	0	2.2288	-2.2288	
16	4.	4.	3.1303	.8697	
17	4.	4.	4.1379	-.1379	
18	6.	6.	5.1659	.3341	
19	8.	8.	6.1098	1.8902	
20	7.	7.	6.8650	.1350	
21	7.	7.	7.3461	-.3461	
22	6.	6.	7.5037	-1.5037	
23	13.	13.	7.3314	5.5686	
24	6.	6.	6.8646	-.8646	
25	4.	4.	6.1705	-2.1705	
26	8.	8.	5.3332	2.6663	
27	4.	4.	4.4387	-.4387	
28	2.	2.	3.5624	-1.5624	
29	3.	3.	2.7505	.2395	
30	1.	1.	2.0678	-1.0678	
31	2.	2.	1.4989	.5011	
32	0	0	1.0526	-1.0526	
33	1.	1.	.7168	.2832	
34	0	0	.4739	-.4739	
35	1.	1.	.3042	.6958	
36	0	0	.1399	-.1899	
37	0	0	.1153	-.1153	
38	0	0	.0692	-.0682	
39	0	0	.0393	-.0393	
40	0	0	.0221	-.0221	
41	0	0	.0121	-.0121	
42	0	0	.0065	-.0065	
43	0	0	.0034	-.0034	
44	0	0	.0017	-.0017	
45	0	0	.0009	-.0009	

After this initial output, the user is asked if actual bucket counts are desired. If 1 (yes) is entered, a table such as the one below is printed with these values. Record clusters can be easily detected from this table and an alternative divisor may be analyzed.

BUCKET COUNT DESIRED? ENTER CODE. 1=YES,2=NO

? 1

NUMBER OF RECORDS HASHED INTO A BUCKET									
BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT	BUCKET COUNT
0	26	1	27	2	23	3	22	4	16
5	28	6	18	7	25	8	27	9	26
10	27	11	20	12	23	13	24	14	25
15	22	16	20	17	22	18	20	19	20
20	22	21	20	22	23	23	17	24	21
25	18	26	24	27	19	28	18	29	21
30	23	31	19	32	35	33	26	34	24
35	19	36	21	37	14	38	20	39	22
40	19	41	21	42	23	43	25	44	19
45	23	46	27	47	23	48	23	49	23
50	33	51	23	52	22	53	23	54	21
55	23	56	21	57	13	58	26	59	26
60	16	61	26	62	17	63	16	64	30
65	16	66	13	67	29	68	17	69	29
70	21	71	29	72	29	73	17	74	18
75	19	76	25	77	26	78	19	79	31
80	19	81	31	82	19	83	26	84	24
85	23	86	24	87	20	88	24		

ENTER CODE. 1=ENTER PARAMS,2=SEARCH MECHANISM ANALYSIS
3=HASHING ANALYSIS,4=IDENTIFIER SET ANALYSIS,5=STOP

6. OPERATING INSTRUCTIONS

To invoke the Search Mechanism Analysis Routine on the MIRJE system at the University of Minnesota, the user should log on the system with an appropriate account number and password,

LOG ON

The BATCH subsystem should then be selected with an RFL memory size of 55000,

/BATCH,55000

and the program is executed by using the procedure file called SMAR.

/-SMAR

The SMAR procedure makes available a file of identifiers, called INFILE, to IDSET. INFILE contains 2000 five-digit random numbers. Each number is in positions 1-5 on a record. To execute the program with a different file of identifiers, say DATA, use:

/-SMAR(INFILE=DATA)